Bending the Future: Autoregressive Modeling of Temporal Knowledge Graphs in Curvature-Variable Hyperbolic Spaces

Jihoon Sohn JIHOONSO@USC.EDU
Department of Mathematics
University of Southern California

Mingyu Derek Ma MA@CS.UCLA.EDU
Department of Computer Science
University of California, Los Angeles

Muhao Chen MUHAOCHE@USC.EDU
Department of Computer Science
University of Southern California

Abstract

Recently there is an increasing scholarly interest in time-varying knowledge graphs, or temporal knowledge graphs (TKG). Previous research suggests diverse approaches to TKG reasoning that uses historical information. However, less attention has been given to the hierarchies within such information at different timestamps. Given that TKG is a sequence of knowledge graphs based on time, the chronology in the sequence derives hierarchies between the graphs. Furthermore, each knowledge graph has its hierarchical level which may differ from one another. To address these hierarchical characteristics in TKG, we propose `HyperVC`, which utilizes hyperbolic space that better encodes the hierarchies than Euclidean space. The chronological hierarchies between knowledge graphs at different timestamps are represented by embedding the knowledge graphs as vectors in a common hyperbolic space. Additionally, diverse hierarchical levels of knowledge graphs are represented by adjusting the curvatures of hyperbolic embeddings of their entities and relations. Experiments on four benchmark datasets show substantial improvements, especially on the datasets with higher hierarchical levels.

1. Introduction

Knowledge graphs (KGs) have been the backbone of many knowledge-driven AI applications [Zhang et al., 2016, Wang et al., 2018, Liu et al., 2018, Huang et al., 2019], where factual knowledge about the real world are described as graphs of entities (nodes) and relations (edges). Yet, facts in real-world are varying over time instead of being persistently unchanged. For example, $\langle JoeBiden, IsPresidentOf, U.S.\rangle$ has been true only for a year-long time, since January 20th, 2021. We do not know when this fact will turn to false–he may run for another presidential term, or he may not–but we know that this will eventually turn to false at some time. Hence, it is meaningful to represent such facts that are dynamically changing over time using a temporal knowledge graph (TKG) in the form of $\langle subject, relation, object, time\rangle$. TKG representation has numerous downstream applications including event prediction [Luo et al., 2020, Deng et al., 2020], transaction recommendation [Ren et al., 2019] and schema induction [Zhang et al., 2020].

The main purpose of TKG reasoning is to forecast future events or facts [Jin et al., 2020, Zhu et al., 2021, Trivedi et al., 2017, 2019]. To precisely define the task, consider a TKG where events lie in a temporal interval $[t_0, t_T]$. Instead of predicting events at timestamps $t \in [t_0, t_T]$, this task, also known as *extrapolation*, aims to predict new facts at a timestamp $t > t_T$. To tackle this task, recent studies have proposed several approaches [Jin et al., 2020, Zhu et al., 2021, He et al., 2021]. Jin et al. [2020] proposed an autoregressive approach while Zhu et al. [2021] utilized the copy-generation mechanism. He et al. [2021] focused on both structural and temporal perspectives. However, there are still a few uncovered challenges. Given that a TKG is a time series of KGs, it is natural that hierarchies can be chronologically derived from TKGs. In particular, one event may evolve into several relevant subevents [Surís et al., 2021], forming hierarchies that represent different paths of evolution among KGs. Little attention has been given to incorporating such hierarchical structures in previous research. Moreover, each KG at one timestamp has different numbers of entities and relations, hence different characteristics as a graph. Note that the characteristics of two graphs at contiguous timestamps are relatively similar compared to those of two graphs at distant timestamps. Previous studies encoded entities and relations in the Euclidean space, which can not fully address optimizing embedding space for each KG of various graph structures. Hyperbolic spaces, often considered as a continuous version of trees, are more advantageous to Euclidean spaces in encoding asymmetric and hierarchical relations [Chami et al., 2019b, Liu et al., 2019]. The curvature of hyperbolic space decides the expanding ratio of the space that fits data structures with a certain exponential factor.

To represent hierarchical and chronological properties of events in the TKG, we propose `HyperVC` (**Hyper**bolic model with **V**ariable **C**urvature), a TKG embedding model in the hyperbolic space instead of a Euclidean spaces. `HyperVC` utilizes hyperbolic spaces in two ways in representations of global information and local information, respectively, of each snapshot of TKG. Global representation summarizes global information of a KG at one timestamp. To represent hierarchical structures among snapshots, all global representations are embedded in the common hyperbolic space. On the other hand, local representations focus on local information such as an entity or an entity-relation pair. To improve the optimization of the embedding space for each snapshot, `HyperVC` gives a variety in the curvature of the embedding space of each snapshot to optimize the embedding spaces that represent KGs of various structures. In other words, distinct structures of KGs are embedded in hyperbolic spaces with different curvatures. Specifically, it is natural for hyperbolic KG embedding models to efficiently represent KGs where the curvature of embedding space was proportional to how hierarchical the graph is [Balazevic et al., 2019, Chami et al., 2020]. In particular, their analyses with Krackhardt hierarchical scores [Krackhardt, 2014] show that the greater the hierarchical score of the data, the better performance at hyperbolic embedding the data showed. Hence, by controlling the curvatures of the embedding space, each embedding space of multiple graphs with different hierarchies can be optimized.

`HyperVC` finds a joint probability distribution of all events in TKG in an autoregressive way, inspired by Jin et al. [2020]. Specifically, to learn global and local representations in hyperbolic spaces, `HyperVC` aggregates the information in the neighborhood using GAT [Veličković et al., 2018], encodes facts using hyperbolic RNN [Ganea et al., 2018a], and decodes as a joint probability distribution of facts. Finally, we infer a curvature of embedding

space in future timestamps using a time series model and predict upcoming events using representations of information of graph at hyperbolic space with inferred curvature.

The technical contributions of this work are as follows: (1) `HyperVC` is the first hyperbolic TKG reasoning method that tackles extrapolation task forecasting future events. (2) Specifically, our method addresses a hierarchy between graphs at different timestamps, which is derived from chronology, and employs it through hyperbolic embedding. (3) Furthermore, `HyperVC` applies hyperbolic RNN to deal with representations in hyperbolic spaces and optimizes the curvatures at each timestamp as time series or functions of the hierarchical scores. (4) `HyperVC` shows a significant improvement in TKG link prediction task, particularly in data with more hierarchical relations.

2. Related Works

Temporal KG reasoning As discussed by Jin et al. [2020], temporal KG reasoning can be divided into two task settings that aim at predicting facts that are positioned differently on the timeline. In the *interpolation* setting, the models [Jiang et al., 2016, Sadeghian et al., 2016, Dasgupta et al., 2018, García-Durán et al., 2018, Leblay and Chekol, 2018, Goel et al., 2020, Montella et al., 2021] infer missing facts at the historical timestamps. To do so, Dasgupta et al. [2018] projected the entities and relations onto timestamp-specific hyperplanes. Leblay and Chekol [2018] and García-Durán et al. [2018] considered the time as a second relation and integrated times with relations.

On the other hand, in the *extrapolation* setting, the models [Jin et al., 2020, Zhu et al., 2021, Li et al., 2021a,b, He et al., 2021, Zhou et al., 2021, Sun et al., 2021, Han et al., 2021] seek to forecast events at unseen (future) timestamp. Jin et al. [2020] defined a joint probability distribution of all facts in an autoregressive function and Zhu et al. [2021] developed a time-aware copy-generation mechanism and applied it in TKG embedding. Li et al. [2021a] and Li et al. [2021b] utilized graph convolutional networks (GCN) to capture structural dependencies between KGs in adjacent timestamps and He et al. [2021] further considered repetitive perspective of relations. Zhou et al. [2021] proposed a framework that is compatible with most sequence models. Sun et al. [2021] proposed a TKG reasoning model that can handle unseen entities and Han et al. [2021] implemented neural ordinary differential equations to forecast future links on TKGs.

Hyperbolic representation learning Data with hierarchical structures can be better represented in the negative-curved hyperbolic space. Theoretically, this is because the circumference of a hyperbolic space grows exponentially with the radius, which aligns with the size growth of hierarchical data that is also exponential with regard to the level of hierarchies. Existing works also support this: Nickel and Kiela [2017] proposed a Riemannian optimization method to learn hyperbolic embeddings supervisedly, and Ganea et al. [2018a] extended neural network operations in the hyperbolic space. Hyperbolic operations in Graph Neural Networks (GNN) using intermediate Euclidean tangent space with differentiable exponential and logarithmic mapping are derived by Liu et al. [2019] and Chami et al. [2019b]. Dai et al. [2021] and Zhang et al. [2021] further introduced a hyperbolic GCN that less relied on Euclidean tangent space. Hyperbolic representation learning has been applied to tasks such as knowledge graph completion [Wang et al., 2021, Balazevic et al., 2019], taxonomy expansion [Ma et al., 2021], organizational chart induction [Chen and Quirk,

2019], event prediction [Surís et al., 2021], classification [López and Strube, 2020, Chen et al., 2020] and knowledge association [Sun et al., 2020]. Most related to our work, Han et al. [2020] proposed DyERNIE to use hyperbolic embeddings to capture geometric features of TKGs. Montella et al. [2021] extended the DyERNIE work and defined the curvature of a Riemannian manifold as the product of both relation and time and shows the helpfulness of the adaptive curvature defined by relations.

While the two hyperbolic TKG embedding methods tackled the interpolation task, our proposed method tackles the extrapolation task of forecasting future events based on the past. In addition to the differences in the targeted tasks, our method is distinctive from the two earlier methods in terms of model setting and definition. Specifically, in Han et al. [2020], the entity representations are set to be "linear" to the time and curvatures are fixed over time. However, `HyperVC` optimizes the entity representations through an auto-regressive way and finds the best curvature at each timestamp. Additionally, Montella et al. [2021] defined the curvature of the Riemannian manifold as a product of two parameters, i.e., the relation-dependent parameter and the time-dependent parameter. While their method may struggle in finding the time-dependent parameter for the future (unseen) timestamp, our method applies to forecasting the future because we train the curvature as a function of times and Krackhardt hierarchical scores.

3. Hyperbolic Spaces

General property A hyperbolic space is a Riemannian space with constant negative curvature, whereas the curvature of a Euclidean space is constantly zero and that of a Spherical space is constantly positive [Iversen and Birger, 1992]. The curvature of a Riemannian space characterizes how the space is locally structured. Particularly, a negative curvature indicates that the volumes grow faster than in the Euclidean space [Cannon et al., 1997].

Two models of hyperbolic spaces Several models describe hyperbolic spaces [Beltrami, 1868, Cannon et al., 1997] and we introduce two models here: the Poincaré ball model and the Lorentz hyperboloid model. The Poincaré ball model, or simply Poincaré model, with curvature $c < 0$, is a d-dimensional ball $\mathbb{B}_c^d = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|^2 < -1/c\}$. The Lorentz hyperboloid model (simply Lorentz model) is another d-dimensional hyperbolic space defined as $\mathbb{L}_c^d = \{\mathbf{x} = (x_0, \cdots, x_d) \in \mathbb{R}^{d+1} | \langle \mathbf{x}, \mathbf{x} \rangle_{\mathbb{L}} = c, x_0 > 0\}$, where c is the curvature. $\mathbb{H}_c^d$ denotes a d-dimensional hyperbolic space of curvature c regardless of models. See Figure 1.

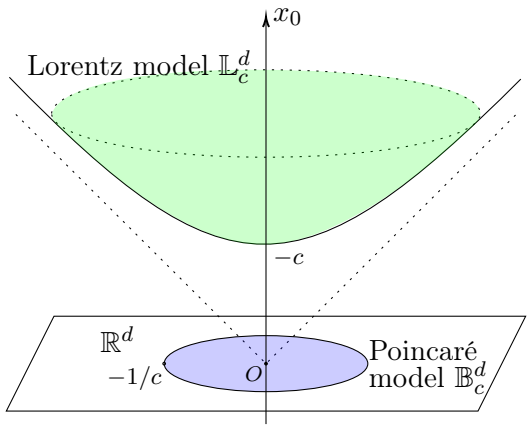

Figure 1: $\mathbb{B}_c^d$ is a Poincaré ball model (inside of purple ball) that are embedded in $\mathbb{R}^{d+1}$ with $x_0 = 0$ and $\mathbb{L}_c^d$ is a Lorentz hyperboloid model (green hyperboloid) of hyperbolic space.

Basic operations: addition and multiplication We introduce addition $\oplus_c$ and multiplication $\otimes_c$ commonly used in neural

networks on hyperbolic spaces. In the Poincaré model, we use Möbius addition and Möbius matrix-vector multiplication. In the Lorentz model, addition and multiplication are performed via the tangent space. For the details, see appendix A.1.

Hyperbolic RNN With the basic hyperbolic operations defined above, we introduce how to generalize a Euclidean RNN to the hyperbolic space [Ganea et al., 2018b]. Traditional RNN is defined as $h_{t+1} = \varphi\left(Wh_t + Ux_t + b\right)$ where φ is a pointwise non-linearity, h_t is the hidden state of previous unit, x_t is the input, and W, U and b are model parameters on Euclidean space $\mathbb{E}$. Given $\mathbb{H}_c$ as the hyperbolic space modeled by a hyperbolic model (such as Poincaré or Lorentz model) with curvature c, $\mathcal{M}$ as a manifold with a flat surface of a certain point locally approximated by the Euclidean space $\mathbb{E}$, we generalize it as follows:

$$h_{t+1} = \varphi^{\otimes_c}\left(W \otimes_c h_t \oplus_c U \otimes_c x_t \oplus_c b\right)$$

where $h_t, x_t, b \in \mathbb{H}_c$ and $W, U \in \mathcal{M}$. Note that the input embeddings are in the hyperbolic space.

4. Method

In this section, we introduce our method `HyperVC`[1].

Notations and problem definitions A TKG contains a set of quadraplets $(s, r, o, t) \in \mathcal{F} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T}$, where $\mathcal{F}$, $\mathcal{E}$, $\mathcal{R}$, $\mathcal{T}$ are the set of valid facts, entities, relations, and timestamps, respectively. G_t denotes the set of facts at timestamp t and thus a TKG can be written as $\{G_t\}_{t\in\mathcal{T}}$. Our goal is to predict a probability $p(s, r, o, t)$ for each triplet (s, r_t, o) of being contained in G_t. Note that among s, r, and o, only the relation r is a time-sensitive variable. To do so, we first assume that G_t depends on previous m snapshot graphs $G_{[t-m,t-1]}$. Inspired by Jin et al. [2020], we decompose the probability $p\left(s, r_t, o|G_{[t-m,t-1]}\right)$ into:

$$p\left(s, r_t, o|G_{[t-m,t-1]}\right) = p\left(s|G_{[t-m,t-1]}\right) \cdot p\left(r_t|s, G_{[t-m,t-1]}\right) \cdot p\left(o|s, r_t, G_{[t-m,t-1]}\right). \quad (1)$$

Namely, when we compute the probability of a triplet (s, r_t, o), we first sample a subject entity s using $p\left(s|G_{[t-m,t-1]}\right)$. Next we calculate the probability $p\left(r_t|s, G_{[t-m,t-1]}\right)$ of r_t given s and the previous m timestamps $G_{[t-m,t-1]}$. Finally we compute the probability $p\left(o|s, r_t, G_{[t-m,t-1]}\right)$ of o given s, r_t, and the previous events $G_{[t-m,t-1]}$.

In the prediction of a missing (future) temporal fact, we infer the missing object entity given $(s, r, ?, t)$ or the missing subject entity given $(?, r, o, t)$. For the former case, the prediction is based on the computation in Equation 1 of probability, and for the latter, we compute the probability similarly as Equation 1 but sampling the object entity o first instead of s.

Model components As shown in Figure 2, `HyperVC` implements *global* representations and *local* representations, as in Jin et al. [2020], to find a joint probability distribution of each event. The global representation $\mathbf{H}_t$ captures the global information of the snapshot G_t at timestamp t, which describes preferences as a graph including trend or periodicity. On the other hand, the local representation $\mathbf{h}_t$ concentrates on local information such as a vertex, an edge, or its neighborhood, thus representing more entity-specific or relation-specific

1. Codes are available at https://github.com/jhsohn11/HyperVC.

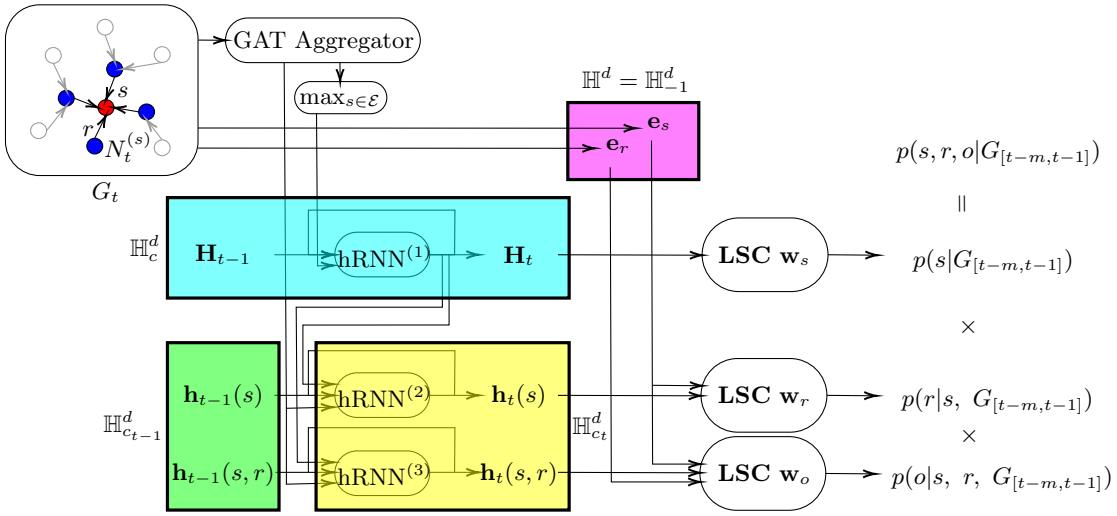

Figure 2: The overall architecture of `HyperVC`. Colored boxes are the hyperbolic spaces that contain the hyperbolic representations with designated curvatures. **LSC w** refers to the linear softmax classifier parametrized by $\mathbf{w}$. Through the cyan box of the global representations, the green and yellow boxes of the local representations, and the pink box of time-consistent hyperbolic embeddings of entities and relations, we calculate the probability of triplet (s, r, o) at the timestamp t.

behaviors. The two representations capture disjoint features of TKGs. Therefore, we utilize both representations to compute the probability distributions of events.

The key idea of `HyperVC` is that both representations $\mathbf{H}_t$ and $\mathbf{h}_t$ are embedded in the hyperbolic space, either with a Poincaré model, or a Lorentz model. We will compare the performance of models embedded in these two hyperbolic spaces in the result part.

Global and local representations First of all, the global representations $\mathbf{H}_t$, $t \in \mathcal{T}$ are all embedded in one hyperbolic space $\mathbb{H}^d_c$ of dimension d with learnable curvature c of hyperbolic space. This is because there is a hierarchy among contiguous snapshots of events derived from chronological order. Specifically, relevant events (for example, events in causal relations) branch into different paths of evolution depending on previously occurred events, and this arises a hierarchy between KGs at neighboring timestamps [Surís et al., 2021]. Hence, we embed the global representations into one common hyperbolic space with curvature c to represent hierarchical structures between different snapshot graphs.

Moving on to the second representation, let $\mathbf{h}_t(s)$ be the local representation in the hyperbolic space $\mathbb{H}^d_{c_t}$ of dimension d with curvature c_t for a subject entity s and $\mathbf{h}_t(s, r) \in \mathbb{H}^d_{c_t}$ be the one for a pair of a subject entity s and a relation r. Unlike the global representation, $\mathbf{h}_t$ has curvatures c_t that vary over timestamp t because each snapshot G_t has a different hierarchical level. For example, one graph G_{t_1} at the timestamp t_1 may have tree-like (hierarchical) structures, which are represented better in hyperbolic space than Euclidean space. For another graph G_{t_2}, they may have less hierarchical relations, whose embedding fits better in Euclidean space. Because the hyperbolic space whose curvature is close to zero is similar to Euclidean space, c_{t_2} will be trained to be closer to zero than c_{t_1}. In this way, we can afford any graphs with diverse hierarchical levels by training the learnable curvature c_t.

These two separate but complementary representations are defined as:

$$\begin{aligned}
\mathbf{H}_t &= \text{hRNN}^{(1)}\left(\mathcal{T}^{c}\left(g'(G_t)\right), \mathbf{H}_{t-1}\right), \\
\mathbf{h}_t(s) &= \text{hRNN}^{(2)}\left(\mathcal{T}^{c_t}\left(g(N_t^{(s)})\right), \mathcal{T}_{c}^{c_t}(\mathbf{H}_t), \mathcal{T}_{c_{t-1}}^{c_t}(\mathbf{h}_{t-1}(s))\right), \\
\mathbf{h}_t(s,r) &= \text{hRNN}^{(3)}\left(\mathcal{T}^{c_t}\left(g(N_t^{(s)})\right), \mathcal{T}_{c}^{c_t}(\mathbf{H}_t), \mathcal{T}_{c_{t-1}}^{c_t}(\mathbf{h}_{t-1}(s,r))\right),
\end{aligned}$$

where hRNN are the hyperbolic RNNs as described in Section 3 [Ganea et al., 2018a], $N_t^{(s)}$ is the subgraph of G_t that contains the entity s, g is the neighborhood aggregator using graph attention [Veličković et al., 2018], and g' is a max-pooling operation defined as $g'(G_t) = \max\left(\{g(N_t^{(s)})\}_{s \in G_t}\right)$ among the aggregated neighborhoods of whole entities s in G_t. While the representations $\mathbf{H}_{t-1}$ and $\mathbf{H}_t$ in $\text{hRNN}^{(1)}$ have the same curvature c, representations in $\text{hRNN}^{(2)}$ and $\text{hRNN}^{(3)}$ have different curvatures. Hence, we adjust the curvatures of $\mathbf{H}_t$ and $\mathbf{h}_{t-1}$ from c and c_{t-1} to c_t through $\mathcal{T}_{c}^{c_t}$ and $\mathcal{T}_{c_{t-1}}^{c_t}$, respectively. Finally, as $g(N_t^{(s)})$ and $g'(G_t)$ are Euclidean vectors, we need a transition $\mathcal{T}^{c_t}$ of these vectors to the hyperbolic space of curvature c_t.

Computations of probabilities Based on both representations, we compute the probability $p(o|s, r_t, G_{[t-m,t-1]})$ as follows:

$$p\left(o|s, r_t, G_{[t-m,t-1]}\right) = \text{LSC}\left(\left[\mathbf{e}_s : \mathbf{e}_{r_t} : \mathcal{T}_{c_{t-1}}^{-1}(\mathbf{h}_{t-1}(s, r_t))\right]^\top \cdot \mathbf{w}_o\right),$$

where $\mathbf{e}_s$, $\mathbf{e}_r \in \mathbb{H}^d$ are learnable hyperbolic representations of the subject entity s and the relation r, respectively, embedded in the hyperbolic space with fixed curvature -1. The local representation $\mathbf{h}_{t-1}(s, r_t)$ collects the information of previous snapshots. However, since $\mathbf{h}_{t-1} \in \mathbb{H}_{c_{t-1}}^d$ while $\mathbf{e}_s$, $\mathbf{e}_{r_t} \in \mathbb{H}_{-1}^d$, we adjust the curvature of $\mathbf{h}_{t-1}$ from c_{t-1} to -1 through $\mathcal{T}_{c_{t-1}}^{-1}$. `HyperVC` tracks and updates the semantic of (s, r) up to t by concatenating both static representations $\mathbf{e}_s$, $\mathbf{e}_r$ and the time-sensitive representation $\mathbf{h}_{t-1}$. Here, we concatenate hyperbolic representations by appending one at the end of another. After the concatenation, `HyperVC` computes the probability of the object entity o by passing a linear softmax classifier LSC parametrized by $\mathbf{w}_o$.

Through the similar processes, we compute the two other probabilities $p\left(s|G_{[t-m,t-1]}\right)$ and $p\left(r_t|s, G_{[t-m,t-1]}\right)$ as follows:

$$\begin{aligned}
p\left(s|G_{[t-m,t-1]}\right) &= \text{LSC}\left(\mathbf{H}_{t-1}^\top \cdot \mathbf{w}_s\right), \\
p\left(r_t|s, G_{[t-m,t-1]}\right) &= \text{LSC}\left(\left[\mathbf{e}_s : \mathcal{T}_{c_{t-1}}^{-1}(\mathbf{h}_{t-1}(s, r_t))\right]^\top \cdot \mathbf{w}_{r_t}\right),
\end{aligned}$$

where the final probabilities are computed by passing a linear softmax classifier parametrized by $\mathbf{w}_s$ and $\mathbf{w}_{r_t}$, respectively.

Learnable curvature Finally, we train the curvature c_t as a function of two variables: times and Krackhardt hierarchical scores. The real-world data such as daily data or weekly data inevitably has a period. As the curvature of the snapshot at each timestamp is affected by how the data look, there exists a seasonal component in the curvature as well. Inspired by Xu et al. [2020], we decompose the curvature as an additive time series

$$c_t = -\sigma\left(\alpha \sin(\omega t) + (\beta t + \gamma)\right), \tag{2}$$

where each term refers to the seasonal component and the trend component. Since the curvature of hyperbolic space is always negative, we take the "Softplus" function σ.

On the other hand, the Krackhardt hierarchical score (the formula is described in appendix A.2) also affects the curvature of the hyperbolic space where the graph is embedded. Then the curvature is represented as

$$c_t = -\sigma(f(Khs_{G_t})), \tag{3}$$

where Khs_{G_t} is the Krackhardt hierarchical score of G_t. To find a function f that best describes the relation between hierarchical score and curvature, we experimented with polynomials: a linear function or a quadratic function.

Finally, we experimented with the combination of these two separate approaches as

$$c_t = -\sigma\left(\alpha \sin(\omega t) + (\beta t + \gamma) + f(Khs_{G_t})\right). \tag{4}$$

The results from these three different approaches were compared in the ablation study in the later section.

Learning objective Given a subject s, a relation r, and a timestamp t, the model predicts the object entity o based on the probability $p(s, r_t, o)$, considering it as a multi-class classification task where each class corresponds to each entity. Then the loss function $\mathcal{L}$ is:

$$\mathcal{L} = -\sum_{(s,r,o,t)\in G} \log p(o|s, r_t) + \lambda \log p(r_t|s)$$

where G is a set of facts, and λ is a hyperparameter that controls each loss term. A similar process works for the subject entity prediction with switched subject and object entities in the loss function.

Inference `HyperVC` predicts future events based on past events. Here, we describe the inference for a missing object, which applies to predicting a missing subject WLOG. Given s, r, t, and the past history of snapshot graphs $G_{[1:t]}$, we predict the object o which has the highest conditional probability. Inspired by Jin et al. [2020], we use multi-step inference over time. `HyperVC` samples the events at the next timestamp based on the conditional probability to build a sample graph and we use this in the inference of future timestamps. In other words, from computing $p(s_{t+1}, r_{t+1}, o_{t+1}|G_{[1,t]})$, or $p(G_{t+1}|G_{[1,t]})$, we get a sample $\widehat{G}_{t+1}$. Then we can further compute $p(G_{t+2}|\widehat{G}_{t+1}, G_{[1,t]})$. Through the iterative computation of conditional distribution and sampling from it, we get $p(G_{t+\Delta t}|\widehat{G}_{[t+1,t+\Delta t-1]}, G_{[1,t]})$, the estimate of $p(G_{t+\Delta t}|G_{[1,t]})$.

5. Experiments

Experimental setup We use four representative TKGs datasets, GDELT [Leetaru and Schrodt, 2013], ICEWS18 [Boschee et al., 2015], WIKI [Leblay and Chekol, 2018], and YAGO [Mahdisoltani et al., 2014]. More details about the datasets can be found in Table 1 and appendix A.3.

We split each dataset into three subsets by train, validation, test with the proportion of approximately 80%, 10%, 10%, respectively, in chronological order. Note that three

subsets contain disjoint timestamps. We report two evaluation metrics for extrapolated link prediction, including Mean Reciprocal Rank (MRR) and H@1/3/10. MRR is the mean of the inverse of ranks of test cases and H@1/3/10 are the proportions of test cases that are ranked in the top 1/3/10, respectively. In the computation of ranks, we use the *filtered* setting, i.e. we filter out the valid triplets that appeared in train, validation, and test sets among candidates.

We compare our `HyperVC` to a diverse set of recent methods for reasoning on static KGs and temporal KGs. Static KG embedding models include TransE [Bordes et al., 2013], DistMult [Yang et al., 2014], ComplEx [Trouillon et al., 2016], ConvE [Dettmers et al., 2018], RotatE [Sun et al., 2019], RGCN-DistMult [Schlichtkrull et al., 2018], and CompGCN-DistMult [Vashishth et al., 2020]. For the static KG methods, we simply remove all the timestamps in datasets and compare the results. On the other hand, temporal KG embedding models include TTransE [Jiang et al., 2016], HyTE [Dasgupta et al., 2018], TA-DistMult [García-Durán et al., 2018], RE-Net [Jin et al., 2020], CyGNet [Zhu et al., 2021], SeDyT [Zhou et al., 2021] and HIP Network [He et al., 2021]. For the other temporal KG embedding models such as CluSTeR [Li et al., 2021a], RE-GCN [Li et al., 2021b], TimeTraveler [Sun et al., 2021], and Tango [Han et al., 2021], they used time-aware evaluation metrics so direct comparison was unavailable.

Dataset	YAGO	WIKI	ICEWS18	GDELT
Entities	10,623	12,544	23,033	7,691
Relations	10	24	256	240
Training	161,540	539,286	373,018	1,734,399
Validation	19,523	67,538	45,995	238,765
Test	20,026	63,110	49,545	305,241
Time gap	1 year	1 year	1 day	15 mins
Timestamps	189	232	304	2,751
Hier. score	0.898	0.976	0.842	0.780

Table 1: Statistics of four datasets. The last row gives the average of Krackhardt hierarchical scores for each dataset.

Results The hyperparameters were searched based on the MRR performance of validation sets. We used the Adam optimizer with a learning rate of 0.001. The batch size was 1024, and the training epoch was (maximum) 100. The loss-controlling hyperparameter λ was set to 0.01. The embedding dimension was tuned among 100, 150, 200, and 300, and the model with dimension 200 mostly outperformed.

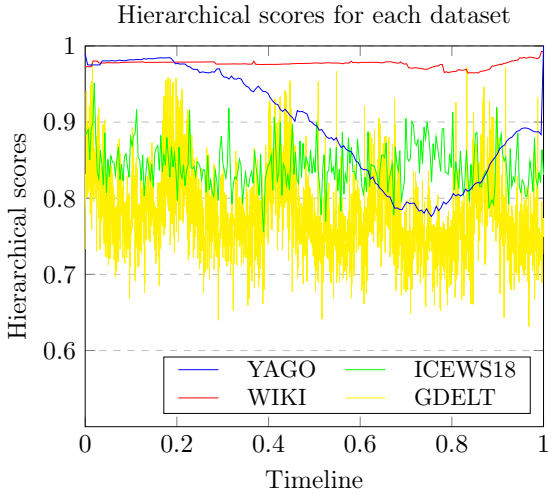

Figure 3: Krackhardt hierarchical stores at each timestamp. The x-axis is the proportioned timestamps compared to the entire dataset.

Table 2 reports the link prediction result by `HyperVC` and other methods on four datasets. The baseline results are adopted from Zhu et al. [2021], Zhou et al. [2021], and He et al. [2021]. On the first two datasets, `HyperVC` outperformed all the previous static and temporal KGs methods except for HIP Network [He et al., 2021]. Especially, `HyperVC` outperformed the state-of-the-arts at H@1 on YAGO. In the WIKI, the most hierarchical dataset among four TKG datasets, `HyperVC` relatively improved the performance

Model	YAGO				WIKI				ICEWS18				GDELT			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE	48.97	46.23	62.45	66.05	46.68	36.19	49.71	51.71	17.56	2.48	26.95	43.87	16.05	0.00	26.10	42.29
DistMult	59.47	52.97	60.91	65.26	46.12	37.24	49.81	51.38	22.16	12.13	26.00	42.18	18.71	11.59	20.05	32.55
ComplEx	61.29	54.88	62.28	66.82	47.84	38.15	50.08	51.39	30.09	21.88	34.15	45.96	22.77	15.77	24.05	36.33
ConvE	62.32	56.19	63.97	65.60	47.57	38.76	50.10	50.53	36.67	28.51	39.80	50.69	35.99	27.05	39.32	49.44
RotatE	65.09	57.13	65.67	66.16	50.67	40.88	50.71	50.88	23.10	14.33	27.61	38.72	22.33	16.68	23.89	32.29
RGCN	41.30	32.56	44.44	52.68	37.57	28.15	39.66	41.90	23.19	16.36	25.34	36.48	23.31	17.24	24.96	34.36
CompGCN	41.42	32.63	44.59	52.81	37.64	28.33	39.87	42.03	23.31	16.52	25.37	36.61	23.46	16.65	25.54	34.58
TTransE	32.57	27.94	43.39	53.37	31.74	22.57	36.25	43.45	8.36	1.94	8.71	21.93	5.52	0.47	5.01	15.27
HyTE	23.16	12.85	45.74	51.94	43.02	34.29	45.12	49.49	7.31	3.10	7.50	14.95	6.37	0.00	6.72	18.63
TA-DistMult	61.72	52.98	63.32	65.19	48.09	38.71	49.51	51.70	28.53	20.30	31.57	44.96	29.35	22.11	31.56	41.39
RE-Net	65.16	63.29	65.63	68.08	51.97	48.01	52.07	53.91	42.93	36.19	45.47	55.80	40.42	32.43	43.30	53.70
CyGNet	63.47	64.26	65.71	68.95	45.50	50.48	50.79	52.80	46.69	40.58	49.82	57.14	50.92	44.53	54.69	60.99
SeDyT-CONV	66.88	−−	67.05	68.73	52.90	−−	52.96	54.00	45.91	−−	45.86	49.54	**54.86**	−−	54.68	58.14
HIP Network	**67.55**	66.32	**68.49**	**70.37**	**54.71**	**53.82**	**54.73**	**56.46**	**48.37**	**43.51**	**51.32**	**58.49**	52.76	**46.35**	**55.31**	**61.87**
`HyperVC`	67.52	**66.46**	67.52	69.28	53.02	51.98	53.36	54.55	41.38	34.21	44.25	55.17	40.08	32.98	42.84	53.26

Table 2: Performances (in percentage) at temporal link prediction task on four datasets. The best results are boldfaced and the second best ones are underlined.

by (maximum) 8.27% (H@1 on WIKI) when compared to RE-Net [Jin et al., 2020], which can be considered as a Euclidean version of `HyperVC`. Our method showed great performances on these datasets because, as we can see in Figure 3 and Table 1, YAGO and WIKI are highly hierarchical data in general.

Experiments further revealed that our approach of implementing hyperbolic spaces strengthened the performance on the datasets with high hierarchical scores (i.e., WIKI, YAGO) whereas, with the other two datasets that have low hierarchical scores (i.e., GDELT and ICEWS18), ours did not outperform the earlier models. Notably, the biggest difference between SoTA (HIP Network [He et al., 2021]) and ours is that the former deploys an additional module to deal with historical vocabulary while ours does not. He et al. [2021] provided the performance scores without the historical vocabulary module on WIKI, and the comparison showed that our model `HyperVC` outperformed SoTA without the module (See Table 3). Furthermore, another model CyGNet [Zhu et al., 2021] which had better performance than ours on the datasets with low hierarchical scores also included a similar module that deals with historical vocabulary. Given this, we speculate that the gap in the performance between ours and the earlier ones may be attributed to the implementation of such historical vocabulary modules. In other words, adding a module of historical vocabulary to our model may improve the performance on these datasets regardless of hierarchical scores.

Model	WIKI			
	MRR	H@1	H@3	H@10
`HyperVC`	**53.02**	**51.98**	**53.36**	**54.55**
HIP w/o hist. module	48.25	39.17	50.36	52.11

Table 3: Comparison of `HyperVC` and HIP Network (SoTA) without the historical vocabulary module.

Table 4 reports the performances of methods that are embedded in two different hyperbolic spaces: `HyperVC` (P) refers to the one embedded in the Poincaré model and `HyperVC` (L) is the one in the Lorentz model. Generally, the model embedded in the Poincaré model shows better performance.

Model	YAGO			WIKI		
	MRR	H@3	H@10	MRR	H@3	H@10
`HyperVC` (P)	**67.52**	**67.52**	**69.28**	**53.02**	**53.36**	**54.55**
`HyperVC` (L)	66.91	67.15	68.62	52.10	52.45	53.90

Table 4: Comparison of `HyperVC` embedded in Poincaré model and `HyperVC` embedded in Lorentz model.

Ablation study As we mentioned, we present an ablation study about the contribution of time-varying curvature of graphs at single timestamps at local representation. We compare four models with different functions that describe the curvature: a learnable constant function, an additive time series as in Equation 2, a function of Krackhardt hierarchical scores as in Equation 3, and a combination of two variables as in Equation 4.

From the results in Table 5, we observe that the model with an additive time series outperformed the other three models on YAGO dataset. The optimized global curvature at YAGO is -2.367, which means the global representation at each timestamp fits better in the hyperbolic space than in Euclidean space. Therefore, as we desired, the hierarchies derived from chronological properties are well represented in the hyperbolic space. On the other hand, the best local curvature at YAGO follows the additive time series model

$$c_t = -\sigma\left[\alpha * t + \beta * \sin(\omega * t)\right],$$

	YAGO			
Model	MRR	H@1	H@3	H@10
HyperVC w/ learnable const.	67.49	**66.46**	67.49	69.12
HyperVC w/ time series	**67.52**	**66.46**	**67.52**	**69.28**
HyperVC w/ hierarchical score	67.49	65.89	67.11	68.61
HyperVC w/ both	66.79	65.57	67.10	68.54
	WIKI			
Model	MRR	H@1	H@3	H@10
HyperVC w/ learnable const.	**53.02**	**51.98**	**53.36**	**54.55**
HyperVC w/ time series	52.51	51.32	52.85	54.10
HyperVC w/ hierarchical score	51.97	50.83	52.30	53.83
HyperVC w/ both	52.16	51.27	52.23	53.68

Table 5: Results of models with different types of learnable curvatures: a constant, a time series, a function of Krackhardt hierarchical scores, and a combination of a time series and hierarchical scores.

where σ is the "Softplus" function, $\alpha = -2.532 * 10^{-2}$, $\beta = -2.846 * 10^{-2}$, $\omega = -6.796 * 10^{-2}$. As α is negative, the curvature gets close to zero as the time increases. Interestingly, the performance got low when Krackhardt hierarchical scores were involved as an independent variable of curvature.

However, as we see in the lower part of Table 5, the model with constant learnable curvature performed better in WIKI. This is because WIKI has rather consistent hierarchical structures compared to YAGO data. (See Figure 3.) Since the hierarchical score of WIKI is steady, the implementing time series or function of the hierarchical score may result in overfitting followed by lower performance.

6. Conclusion

In this paper, we proposed HyperVC to tackle the extrapolation task on TKG reasoning in an autoregressive way. To address hierarchical relations within TKG, we relied on hyperbolic space rather than Euclidean space in both global and local representations. The global representation elaborates the hierarchies between knowledge graphs at different timestamps while local representation captures diverse hierarchical levels of knowledge graphs through the learnable curvature. According to the experimental results, HyperVC performs great in the link prediction task on more hierarchical datasets.

Acknowledgements

The authors appreciate the reviewers and editors for their insightful comments and suggestions. Muhao Chen is supported by the National Science Foundation of United States grant IIS 2105329 and a faculty research award from Cisco.

References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. Multi-relational poincaré graph embeddings. *Advances in Neural Information Processing Systems*, 32:4463–4473, 2019.

Eugenio Beltrami. Teoria fondamentale degli spazii di curvatura costante. *Annali di Matematica Pura ed Applicata (1867-1897)*, 2(1):232–255, 1868.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. Icews coded event data. *Harvard Dataverse*, 12, 2015.

James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 31:59–115, 1997.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4869–4880, 2019a.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019b.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Boli Chen, Xin Huang, Lin Xiao, Zixin Cai, and Liping Jing. Hyperbolic interaction model for hierarchical multi-label classification. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7496–7503, 2020.

Muhao Chen and Chris Quirk. Embedding edge-attributed relational hierarchies. In Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer, editors, *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 873–876, 2019.

Jindou Dai, Yuwei Wu, Zhi Gao, and Yunde Jia. A hyperbolic-to-hyperbolic graph convolutional network. 2021.

Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2001–2011, 2018.

Songgaojun Deng, Huzefa Rangwala, and Yue Ning. Dynamic knowledge graph based multi-event forecasting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1585–1595, 2020.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018a.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5350–5360, 2018b.

Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4816–4821, 2018.

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3988–3995, 2020.

Çaglar Gülçehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter W. Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. Hyperbolic attention networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. Dyernie: Dynamic evolution of riemannian manifold embeddings for temporal knowledge graph completion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7301–7316, 2020.

Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8352–8364, 2021.

Yongquan He, Peng Zhang, Luchen Liu, Qi Liang, Wenyuan Zhang, and Chuang Zhang. Hip network: Historical information passing network for extrapolation reasoning on temporal knowledge graph. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1915–1921, 2021.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. Knowledge graph embedding based question answering. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 105–113, 2019.

Birger Iversen and Iversen Birger. *Hyperbolic geometry*, volume 25. 1992.

Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Baobao Chang, Sujian Li, and Zhifang Sui. Towards time-aware knowledge graph completion. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1715–1724, 2016.

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6669–6683. Association for Computational Linguistics, 2020.

David Krackhardt. Graph theoretical dimensions of informal organizations. In *Computational organization theory*, pages 107–130. Psychology Press, 2014.

Julien Leblay and Melisachew Wudage Chekol. Deriving validity time in knowledge graph. In *Companion Proceedings of the The Web Conference 2018*, pages 1771–1776, 2018.

Kalev Leetaru and Philip A Schrodt. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer, 2013.

Zixuan Li, Xiaolong Jin, Saiping Guan, Wei Li, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. Search from history and reason for future: Two-stage reasoning on temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4732–4743, 2021a.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 408–417, 2021b.

Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2395–2405, 2018.

Federico López and Michael Strube. A fully hyperbolic neural model for hierarchical multi-class classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 460–475, 2020.

Wenjuan Luo, Han Zhang, Xiaodi Yang, Lin Bo, Xiaoqing Yang, Zang Li, Xiaohu Qie, and Jieping Ye. Dynamic heterogeneous graph neural network for real-time event prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3213–3223, 2020.

Mingyu Derek Ma, Muhao Chen, Te-Lin Wu, and Nanyun Peng. HyperExpan: Taxonomy expansion with hyperbolic representation learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4182–4194, 2021.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *7th biennial conference on innovative data systems research*. CIDR Conference, 2014.

Sebastien Montella, Lina M. Rojas Barahona, and Johannes Heinecke. Hyperbolic temporal knowledge graph embeddings with relational and time curvatures. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3296–3308, 2021.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6338–6347, 2017.

Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten De Rijke. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4806–4813, 2019.

Ali Sadeghian, Miguel Rodriguez, Daisy Zhe Wang, and Anthony Colas. Temporal reasoning over event knowledge graphs. In *Workshop on Knowledge Base Construction, Reasoning and Mining*, 2016.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *The Semantic Web*, pages 593–607. Springer International Publishing, 2018.

Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. TimeTraveler: Reinforcement learning for temporal knowledge graph forecasting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8306–8319, 2021.

Zequn Sun, Muhao Chen, Wei Hu, Chengming Wang, Jian Dai, and Wei Zhang. Knowledge association with hyperbolic knowledge graph embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5704–5716, 2020.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019.

Dídac Surís, Ruoshi Liu, and Carl Vondrick. Learning the predictability of the future. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12607–12617, 2021.

Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *international conference on machine learning*, pages 3462–3471. PMLR, 2017.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*, 2019.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.

Abraham A Ungar. Hyperbolic trigonometry and its application in the poincaré ball model of hyperbolic geometry. *Computers & Mathematics with Applications*, 41:135–147, 2001.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*, 2020.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*, pages 1835–1844, 2018.

Shen Wang, Xiaokai Wei, Cicero Nogueira Nogueira dos Santos, Zhiguo Wang, Ramesh Nallapati, Andrew Arnold, Bing Xiang, Philip S Yu, and Isabel F Cruz. Mixed-curvature multi-relational graph neural network for knowledge graph completion. In *Proceedings of the Web Conference 2021*, pages 1761–1771, 2021.

Chenjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Yazdi, and Jens Lehmann. Temporal knowledge graph completion based on time series gaussian embedding. In *The Semantic Web – ISWC 2020*, pages 654–671, Cham, 2020. Springer International Publishing. ISBN 978-3-030-62419-4.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362, 2016.

Hongming Zhang, Muhao Chen, Haoyu Wang, Yangqiu Song, and Dan Roth. Analogous process structure induction for sub-event sequence prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1541–1550, 2020.

Yiding Zhang, Xiao Wang, Chuan Shi, Nian Liu, and Guojie Song. Lorentzian graph convolutional networks. *ArXiv preprint*, abs/2104.07477, 2021.

Hongkuan Zhou, James Orme-Rogers, Rajgopal Kannan, and Viktor Prasanna. Sedyt: A general framework for multi-step event forecasting via sequence modeling on dynamic entity embeddings. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3667–3671, 2021.

Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4732–4740, 2021.

Appendix A. Appendix

A.1 Basic Operations: Addition and Multiplication

We first introduce basic addition and multiplication operations commonly used in neural networks for both Poincaré and Lorentz models. Unlike in a Euclidean space, the addition of two vertex vectors in hyperbolic space is different from axis-wise addition. In Poincaré model, we use Möbius addition $\oplus_c$ for $\mathbf{x}, \mathbf{y} \in \mathbb{B}$ follows

$$\mathbf{x} \oplus_c \mathbf{y} := \frac{(1 - 2c\mathbf{x} \cdot \mathbf{y} - c\|\mathbf{y}\|^2)\mathbf{x} + (1 + c\|\mathbf{x}\|^2)\mathbf{y}}{1 - 2c\mathbf{x} \cdot \mathbf{y} + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}$$

for addition operation where $\cdot$ is the dot product of two vectors [Ungar, 2001, Ganea et al., 2018b, Gülçehre et al., 2019]. Note that as c goes to 0, the Möbius addition converges to normal addition in Euclidean space. For multiplication, Möbius matrix-vector multiplication $\otimes_c$ is defined as

$$M \otimes_c \mathbf{x} := (1/\sqrt{c}) \tanh\left(\frac{\|M\mathbf{x}\|}{\|\mathbf{x}\|} \tanh^{-1}(\sqrt{c}\|\mathbf{x}\|)\right) \frac{M\mathbf{x}}{\|M\mathbf{x}\|}.$$

In the Lorentz model, we perform addition and multiplication via the tangent space. The logarithmic operation transforms vectors from $\mathbb{L}_c^d$ to the Euclidean tangent space $\mathbf{T}_x\mathbb{L}_c^d$ associated with the point $x \in \mathbb{L}_c^d$ and exponential operation conducts the reversed transformation [Ma et al., 2021]. Given $c = -1/K(K > 0)$, $\langle ., . \rangle_{\mathcal{L}}$ as the Minkowski inner product and $d_{\mathcal{L}}^K(x, y) = \sqrt{K} \operatorname{arcosh}\left(-\langle x, y\rangle_{\mathcal{L}}/K\right)$, they are defined as:

$$\exp_x^K(v) = \cosh\left(\frac{\|v\|_{\mathcal{L}}}{\sqrt{K}}\right) x + \sqrt{K} \sinh\left(\frac{\|v\|_{\mathcal{L}}}{\sqrt{K}}\right) \frac{v}{\|v\|_{\mathcal{L}}}, \log_x^K(y) = d_{\mathcal{L}}^K(x, y) \frac{y + \frac{1}{K}\langle x, y\rangle_{\mathcal{L}} x}{\left\|y + \frac{1}{K}\langle x, y\rangle_{\mathcal{L}} x\right\|_{\mathcal{L}}}.$$

Hence, we could define matrix addition and multiplication on the Lorentz model by setting x to to the origin point $\mathbf{o}$ if $P_{\mathbf{o} \to \mathbf{x}}^K(\cdot)$ is the parallel transport from $\mathbf{T}_{\mathbf{o}}\mathbb{L}_c^d$ to $\mathbf{T}_x\mathbb{L}_c^d$ [Chami et al., 2019a]:

$$M \otimes^K \mathbf{x} := \exp_{\mathbf{o}}^K\left(M \log_{\mathbf{o}}^K(\mathbf{x})\right), \mathbf{x} \oplus^K \mathbf{y} := \exp_{\mathbf{x}}^K\left(P_{\mathbf{o} \to \mathbf{x}}^K(\mathbf{y})\right).$$

A.2 Krackhardt Hierarchical Scores

The Krackhardt hierarchical score measures how hierarchical the graph is as follows:

$$Khs_G = \frac{\sum_{i,j=1}^n R_{i,j}(1 - R_{j,i})}{\sum_{i,j=1}^n R_{i,j}},$$

where R is the adjacency matrix, i.e. $R_{i,j} = 1$ if there is an edge from node i to j and 0 otherwise. For example, a graph full of symmetric relations has $Khs_G = 0$ while a tree (a graph with no symmetric relation) has $Khs = 1$. See Krackhardt [2014] for more details.

A.3 TKGs Datasets

The first dataset is a part of the dataset Global Databases of Events, Language, and Tone, or GDELT, and collected from 1/1/2018 to 1/31/2018 with a time interval of 15 minutes. ICEWS18 is from daily-event-based TKG Integrated Crisis Early Warning System (ICEWS), extracted from 1/1/2018 to 10/31/2018. The last two datasets are subsets of Wikipedia history and YAGO3, respectively. They both contain temporal fact with time frames in the form of $(s, r, o, [t_s, t_e])$, where t_s is the starting time and t_e is the ending time. Following the prior works [Jin et al., 2020, Zhu et al., 2021], we divide the temporal facts (WIKI and YAGO) into timestamps with a time interval of one year.